Vol.7 Nº2.   October 1993.

FORMAT

FOR SPECTRUM AND SAM USERS

-JON-
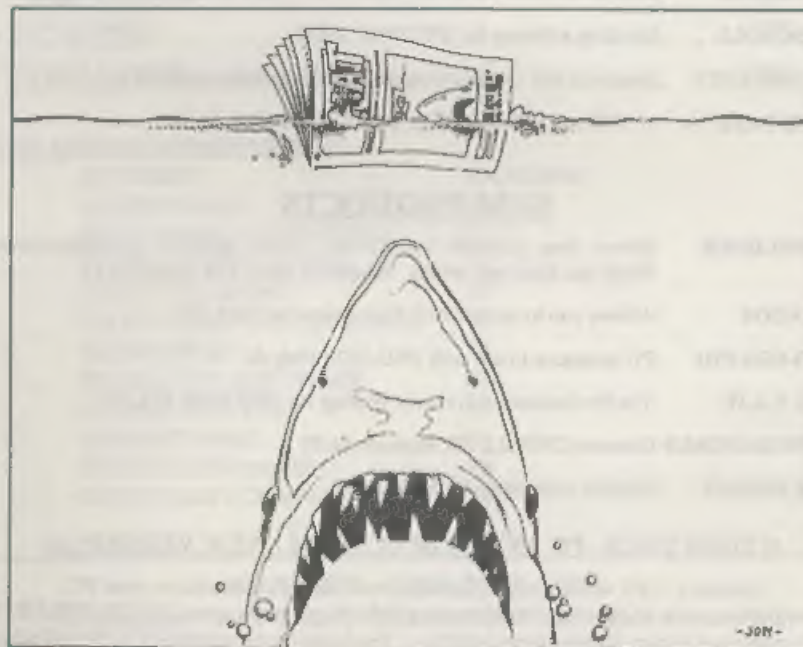
Packs More Byte
Than A Killer White

£1.25 (UK R.R.P.)   ISSN 0963-8598.

B.G.SERVICES

CONTENTS

Vol.7 Nº2.   **October 1993.**

NEWS ON 4

MEMORY CHIP SHORTAGE

The computer world is in the grip of yet another memory shortage - the third major one in the last five years.

The new excuse? *A fire at a far-east epoxy resin plant.* The fact that experts believe that there is at least a six year stock-pile in the hands of distributors and manufactures seems to have been ignored by those who want to hike the prices of RAM yet again.

West Coast Computers have been having major difficulty getting stocks of RAM to build SAMs and are currently paying over 60% more than they did in January. At least one supplier even took the order (and the money) and then turned round and said "Oh sorry, our stocks have just been sold to one of our big customers, we will have some more for you in the next 10 days or so but the price may be a lot higher."

West Coast have promised to hold down prices for the moment - but if things continue like this for long then items using memory chips will almost certainly have to go up.

SINCLAIR BACK IN COMPUTERS

No, sorry to say it is not Uncle Clive, but his son Crispin who has just formed a new company 'Sinclair Direct'.

The company will be selling games for the IBM compatible market when mail order advertising starts in mid-October.

NEW PD LIBRARY FOR SAM

Derrick Morgan, who has been involved with the SAM Coupé right from the start, is launching a new PD software library. Over the last two years

many PD libraries have been set up, mostly by quite young individuals, only to fail after a few months. With Derrick's experience (one could even say maturity) the new venture is sure to stand the test of time.

To start with, Derrick is looking for as much SAM PD as he can get hold of. Once he has had time to test, catalogue and grade the software he will start running adverts in FORMAT so SAM users can get hold of the software.

Anyone with any PD software, both new and old, who wants to contribute to the library can send a copy to Derrick Morgan at 18 Mill Lane, Glenburn Road, Old Skelmersdale, Lancs, WN8 8RH. If you include a stamped addressed envelope he will send you your disc back.

SPECTRUM BOOKS

A limited number of Spectrum books are still available from one supplier.

Computer Manuals of Birmingham have stocks of '30 Hour Basic - Spectrum Edition', 'Mastering Machine Code' (the one by Tony Baker), 'Step By Step Programming (books 1 and 2)' and 'Step By Step Graphics (again books 1 & 2)'.

Apart from the Spectrum books there are a vast number of other computer titles listed - they claim over 3500 titles in stock

Send an A4 size SAE to Computer Manuals, Freepost, 50 James Road, Birmingham. They will send you their latest catalogue (the Spectrum books are on page 30 by the way).

The Editor Speaks

I'm sorry to say this month has been a bit if a disaster. I've spent nearly a fortnight off with bronchitis (if I could find the rat that gave me the cold that started it....) so I'm way behind with everything - especially *FORMAT*, hence the slightly smaller issue. It also meant that many members who tried to reach the hotline kept getting the impenetrable wall of the dreaded answering machine, sorry, but even if I had been here I'm sure the germs were potent enough to travel down the phone lines and infect others.

Still, at least Jenny survived, without infection, to keep things running.

My thanks to all of you who took the time to comment on the new layout, I was really proud when last months issue went out - the hard work was worth it in the end. There are still some changes I plan to make but nothing drastic in the near future. It also means that there are still several articles not transferred from the old system - the hardware course, for instance, will be back next month.

With the new technology installed we could now cope with a bigger issue size but that will depend on advertising revenue - on average one page of advertising allows me to print two extra pages of text. The problem now is that we are close to the limit that can be posted at normal letter rate - there is then a massive 55% increase in the cost of postage if we print a larger issue. I hope that one day, soon, a 60 page issue (or even larger) will be possible - and if we can raise the number of advertisers we will do it.

Next, the latest on the Gloucester Gathering [*and no, despite what one* joker *suggested in a letter, this is not a Scottish gathering of the clans - so kilts are not compulsory attire*] set for Saturday 13th November. There will be a map in the November issue which will go out in time to reach you about a week before the show. Tickets (also including a map) will go out late October so don't worry if you have sent off for some. It is not an all-ticket affair but the space in the hall is limited and priority will be given to ticket holders (and anyway, if you order your tickets in advance you save money). Remember tickets will cost you £2 on the door or £1.50 if ordered in advance. There is also a family ticket (for two adults and any number of kids [under 16s]) priced at £5 (£4 in advance). To order tickets send a cheque and an SAE for your tickets to be returned in.

While there will be as many trade stands as we can fit in, although this is primarily a get-together for Spectrum and Sam owners. Carol Brooksbank will be there, Nev Young hopes to make it and Adrian Parker from Blue Alpha has promised to put in an appearance. The hope is that there will be several demonstrations going on through the day and lots of interesting discussion groups forming. If you have anything you could demo to people give me a ring and we will see what we can do. It promises to be a day not to be missed by any true Spectrum/Sam enthusiast so get your orders for tickets off straight away.

That's about it for this month, November's issue will be jammed with goodies, so don't miss it.

Until next month.

Bob Brenchley, Editor.

SHORT·SPOT
YOUR HINTS, TIPS AND PROGRAMMING IDEAS

Edited By:- John Wase.

Short Spot time again, folks, and this month we kick off right away with a little contribution from John Redfern of Forest Hill, London. John has sent me a useful little routine for those with a Spectrum and another different one for those with a SAM. Though it's better if you've got both...

Suppose you have. You know what a nuisance it is to have a SAMprog that you've laboriously adapted to the Spectrum, but which won't autoload any more. Well, John has the answer - here are two programs which change the AUTO* autoload command within SAMDOS or MASTERDOS to an alternative name, enabling you to create dual format discs which will Autorun both on SAM and on the DISCiPLE/PLUS D.

John writes that the AUTO* filename within both MASTERDOS and SAMDOS can be readily changed with the following POKEs, in each case *newname* must be 10 characters long including any spaces.

SAMDOS:-
```
POKE DVAR 7572,"newname "
```
MASTERDOS:-
```
POKE DVAR 9144,"newname "
```

The program "PokeDVAR" does this automatically, making sure the length is right.

```
 10 CLS #: MODE 4: POKE SVAR 6
    18,8
 20 INPUT #2"NEW AUTOLOAD NAME
    "n$
 30 IF LEN n$<10 THEN LET n$=n
    $+" ": GO TO 30
 40 IF LEN n$>10 THEN PRINT "N
    ame too long": GOTO 20
 50 PRINT '"Press 1 for SAMDOS
    "
 60 PRINT "Press 2 for MASTERD
    OS"
 70 GET d
 80 GOTO ON d;90,100
 90 POKE DVAR 7572,n$: GOTO 10
    5
100 POKE DVAR 9144,n$: GOTO 10
    5
105 PRINT '"NEW AUTOLOAD NAME
    "; PAPER 10;n$: STOP
200 DEF PROC aa: SAVE "POKEDVA
    R": ENDPROC
```

The other program is "PeekAUTO". This automatically locates the page that DOS is loaded at and the start address, then displays the information, prompts you for a new Filename, and then detects whether you are using SAM Basic, MASTERBASIC, SAMDOS or MASTER-DOS, and in what combination. It also tells you if the SAM is 256 or 512k and finally it saves the correct DOS file. The disc you are saving on must have position 1 free, as the file that is saved may not have the same name you usually use.

```
10 REM +++++ PEEKAUTO +++++
20 REM ++ AUTO* RENAME ++
       by John Redfern 1993.
30 MODE 4: WINDOW : SCROLL CL
   EAR
40 PAPER 9: BORDER 9: PEN 15:
   CLS : CSIZE 8,9: POKE SVAR
   2,63: POKE SVAR 618,8: LET
   adr=1
50 INPUT "NAME TO FIND ";name
   $
```

```
 60 IF LEN name$>10 THEN PRINT
    "Name too long": GOTO 50
 70 IF LEN name$<10 THEN LET n
    ame$=name$+" ": GOTO 70
 80 LET dosadr=PEEK SVAR 450*1
    6384+16384: REM find DOS a
    ddress
 90 LET page=PEEK SVAR 450+1:
    REM find DOS page
100 IF PEEK SVAR 692=31 THEN L
    ET start=16384*page,end=st
    art+16384: ELSE LET start=
    16384*page,end=start+16384
110 LET adr= INSTR(adr,MEM$(st
    art TO end),name$)
120 PRINT "start = ";start,"en
    d = ";end' "adr  = ";adr,
130 IF adr<>0 THEN LET
    addr=start+adr-1
140 PRINT "addr = ";addr
150 PRINT ' PEN 15; PAPER 10;n
    ame$;: PRINT "found at:-"
160 FOR n=addr TO addr+9
170 LET z=PEEK n
180 PRINT n;" = ";z;,CHR$ z
190 NEXT n
200 WINDOW 0,31,15,18
210 INPUT #2"New Name "newname
    $: IF newname$="ZZZ" THEN
    STOP
220 IF LEN newname$>10 THEN PR
    INT "Name too long": GOTO
    210
230 IF LEN newname$<10 THEN LE
    T newname$=newname$+" ": G
    OTO 230
240 POKE addr,newname$
250 PRINT PEN 15; PAPER 10; ne
    wname$: PRINT "is new AUTO
    LOAD filename"
260 PRINT "Press a Key to Resa
    ve DOS"
270 PAUSE
280 REM Determine DOS/BASIC
    configuration & SAVE it
290 LET md=0
300 FOR n=20736 TO 20767: IF P
    EEK n=48 THEN LET md=1
310 NEXT n
320 PRINT "SAMDOS " AND PEEK D
    VAR 7<22;"MASTERDOS " AND
    PEEK DVAR 7>29;"+ MASTERBA
    SIC " AND md=1;"256K" AND
    PEEK SVAR 692=15;"512K" AN
    D PEEK SVAR 692=31
330 REM The next two lines can
    only be ENTERED if using
    MASTERBASIC
340 IF md=1 AND PEEK DVAR 7<22
```

```
    THEN SAVE BOOT "sd+mbas":
    GOTO 380
350 IF md=1 AND PEEK DVAR 7>29
    THEN SAVE BOOT "md+mbas":
    GOTO 380
360 IF PEEK DVAR 7<22 THEN SAV
    E OVER "SAMDOS2"CODE dosad
    r+9,10000: GOTO 380
370 IF PEEK DVAR 7>29 THEN POK
    E 65536,MEM$(dosadr TO dos
    adr+15700): SAVE OVER "mdo
    s"CODE 65536,15700: GOTO 3
    80
380 WINDOW : SCROLL RESTORE :
    STOP
390 DEF PROC SA: SAVE "PEEKAUT
    O" LINE 1: END PROC
```

Finally, John mentions that PeekAUTO will also find the name of the current program in memory at address 491835. John is sure that this must be of some use to someone, but hasn't thought of a use for himself yet.

Fine. We've a lot of brains among *FORMAT* readers. Anyone else think of a use?

Many thanks, John.

Next, a little piece from a squiggle that looks like John Saunders of Chalfont St Giles, Bucks. I hope that's right, John! John mentions the COLOUR aspects of SAM's abilities, and while he is not rash enough to imagine that someone has a full and exhaustive knowledge of SAM's quirks over line interrupts, he is sure that Roy Gardner's multicolour program has started off some more hares. John has a plea - "Where does all the gen come from? Please explain to us dunderheads about these strange beasts!" Sorry, John; I'm joining you. I'm a dunderhead, too.

Anyway, let's look at some of John's explanations, and then perhaps someone will extend them a bit in future issues. Don't forget: unless you put it in Short Spot, no-one else will know, and that will be a pity.

John explains things like this. "I

THINK," he writes, "that they are the brief intervals that the computer has between dealing with the scanning of the TV lines - do they correspond with the 'plotted pixels' lines? I have been trying to tinker round with the principles that Roy's program seems to imply - the strange line:-

    40 PALETTE 1, c LINE y

for instance."

He goes on to say, "The only reference to 'LINE' that I have found in this sort of context is in the MasterBasic User Guide, p.28 and that is not compatible, logically, with 'y=172'!"

John has clearly done some more burrowing at this stage, for in ink is added "Sorry - yes it is!"

The following program took John a short distance along the line, but further burrowing suggested looming dead-ends and cul-de-sacs.

```
10 LET L=172, COL=40
12 FOR N=0 TO 18
15 PALETTE 1, COL LINE K
17 PALETTE 2, COL+35 LINE K-1
   REM COL+ your choice
20 PRINT AT N,0; PEN 1;"******
   *+++#######"; PEN 2; "******
   ***+++$$$$$2: LET K=K-7, C
   OL=COL+3
25 NEXT N
30 REM
40 REM
50 REM
60 REM
70 REM
80 REM
```

"What is now interesting is that the Direct Command of PEN 2 (or 1), followed by LIST gives a very pretty listing - or directory with DIR 1, and an interesting SAVEing display".

Finally, John asks me to "Do a short (long?!) piece on Line Interrupts."

John; I am glad the column is enjoyed so much, but as for my writing a short or

long piece about line interrupts; well, I haven't the technical knowledge. But I bet someone out there HAS. Can anyone help further?

Next, we have something from Carol. Carol? Yes; Carol Brooksbank, who sends a little tip about *Style Writer*. She discovered recently that there is a bug in the multiple copy printing routine. If you ask for, say, ten copies, all you get is one copy - and nine sheets of blank paper! This is because the variables set up at the printing menu are not all preserved as they should be by the printing routine, and after the first copy, all you get are form feeds. Oh dear! But don't despair; here's Carol's bug-fix. Load the program (it doesn't matter if you use the colour or monochrome version). Use CTRL/B to exit to Basic. Now edit line 1700 so that it reads:-

```
1700 IF q=201 THEN delfunc:mess
     age"PLEASE WAIT":lastline:
     INPUT:prmenu: LET xprin=PR
     IN,xlast=LAST,xpic$=PICTUR
     E$,xnl$=NLQ$: FOR cop1 TO
     copies: LET PRIN=xprin,LAS
     T=xlast,PICTURE$=xpic$,NLQ
     $=xnl$:prfile2: NEXT cop:i
     nsfunc
```

When you are in Basic, you can, if you like, customize the printer codes found in the DATA statements between lines 3200 and 3360 so that the default codes on loading the program will be the ones you need. You have four codes in a sequence, and if your sequence needs less than four numbers, you must fill in the sequence with -1 for each unused one. For instance, the codes 27, 67, 70, -1 are for a full A4 page length. Now put your working disc of STYLE WRITER in drive 1 and, still in Basic, type RS and the customised copy will be saved.

Many thanks Carol; that's smashing.

Now over to the Spectrum again. D'you

run BetaDOS? Miles Kinloch of Edinburgh (funny, that name seems familiar) does. And he's found a bug, too. And fixed it. Here's what Miles says...

"The bug concerns the special version of the FORMAT command used to format a disc for more than 80 catalogue entries, e.g. FORMAT d1,10 etc. The drive to be formatted will in fact be the last one which was used, rather than the drive specified in the command. Potentially this could have serious effects, leading to valuable discs being wiped in error, so I thought it would be well worth fixing. Here's the cure I've come up with.

```
10 REM     BETADOS BUG FIX
20 REM     By M. Kinloch
30 REM
40 CLEAR 4e4: LOAD d1"+sysBeta
   "CODE 40960
50 FOR a=46403 TO 46410: READ
   d: POKE a,d: NEXT a
60 DATA 215,130,28,247,200,215
   ,153,30
70 PRINT "PRESS ANY KEY TO RES
   AVE BETADOS": PAUSE 0
80 SAVE d1"+sysBeta"CODE 40960
   ,6850
```

Many thanks, Miles.

Next a crisp little tip from D.W.Stokes of Launceston, Cornwall, who writes that he has a Brother HR5 which uses a special ribbon, but can also use thermal paper, and Fax paper works just fine in this. Here's a sample of thermal underwear - oops - Faxpaper output. Looks fine to me.

```
ESC Q n
Print width 1 columns--Print
-----Print width 4 columns---
Print width 6 columns--------
Print width 8 columns--------
```

As for the Alphacom; the original used special blue paper. But you can get rolls from Tandy quite cheaply which print black and, so Mr Stokes says, "look better than the blue Alphacom stuff." Many thanks.

Funnily enough, the next letter I opened is from Alan Cox. Printed on Fax paper on a Brother HR5. He also asks if I have a general policy on length. Well, Alan; I prefer 'em short. Unfortunately, my time is short, also. When I get a big pile of envelopes and open them and find them full of unsuitable material, and I've got Bob bawling, saying "Gotta send it in tonight's post: going to the printer tomorrow", then my index of desperation rises sharply. Like the young man from Liverpool, from whom I've just received some half dozen pages of handwritten programs, and I just haven't the time to type them in. So the length of the items bears a direct mathematical relationship to my index of desperation, if you see what I mean. I think it's more logarithmic than linear, though!

Even so, I am asking for 'Merge' items in an attempt to keep things short and introduce a fair variety.

Now back to the Spectrum, and Lee Willis's 'Quirk' mentioned in the September issue. Ettrick Thomson writes that the Spectrum behaves in just the same way, so the explanation is to be found in Logan and O'Hara's 'Spectrum ROM Disassembly'. Unfortunately, to understand this, you have to wade through six and a half pages of machine code, albeit annotated. The particular point which leads to such unexpected results is that the command DRAW x,y,z draws an arc of a circle, covering z radians and passing through two particular points. If z>2*Pi, a complete circle will be drawn, perhaps more than

once. With z=501 (about 160*Pi), the circle is drawn about 80 times. Now 'drawn' means, as is usual with curves, a sequence of straight lines, usually short enough to give the impression of a curve. In DRAW x,y,z however, the number of straight lines is limited to 252, which is the case with z=501. It is this limit which provides the odd behaviour, for 250 lines going 80 times round and round means only about three lines per revolution. If it had been exactly three, the circle would have been an equilateral triangle, repeated about 80 times.

This lines up with what you see when you run 'Quirk'. A series of 252 equal-length straight lines, each making an angle roughly sixty degrees with its predecessor, their ends lying on a circle. You get similar results with values other than 40,40,501 for x,y,z so long as z is large enough. The commands PLOT 128,40: DRAW 40,40,529 show the near-equilateral triangle more closely.

Many thanks, Ettrick.

This business trying to do things which the originators of the algorithms hadn't thought of your doing is fascinating. Gilbert Jackson of Cregiau, Cardiff, sends this little Spirograph jobbie which, he says, uses the device featured in LINE 120, and suggested by Ettrick to avoid the cumulative errors in Barry Twyman's original program, and also Alan Cox's modification. In essence, it simulates the DRAW TO x,y provided in BetaBasic and SAMBasic, and therefore avoids cumulative errors. The circles drawn by LINE 46 are not proper to the program itself and can be omitted, but Gilbert thinks they throw some light on the behaviour of the pen as it is moved from the edge of the wheel (d=0) towards its centre (d=1). The outer circle is, of course, the ring itself.

The program does not cater for the wheel outside the ring, but something like that can be seen if the second SIN in LINE 100 is made COS, and the second COS in line 110 is made SIN. The inner circle of LINE 46 then represents the ring, despite the text in LINE 20.

```
3 REM For Spectrum+ 48k
  Barry Twyman's Spirograph
  (FORMAT 6/93) plus some of
  Alan Cox 9/93 with Ettrick
  Thomson's Line 120
4 PAPER 0: BORDER 0: PEN 7
10 CLS
20 INPUT "radius of outer ring
   ? (r<=87)";r: PRINT "r=";r
25 IF r<1 OR r>87 THEN GOTO 10
30 INPUT "radius of wheel? (r1
   <=r) ";r1: PRINT "r1=";r1
40 INPUT "How far pen from edg
   e of wheel? (as a fraction
   0 to 1) ";d: PRINT "d=";d
45 LET d=d*r1
46 CIRCLE 127,87,r: CIRCLE 127
   ,87,r-2*r1: REM optional
50 LET s=PI/20
60 LET a=0
100 LET x=(r-r1)*SIN a+(r1-d)*S
   IN (a*(1-r/r1))+127
110 LET y=(r-r1)*COS a+COS (a*(
   1-r/r1))*(r1-d)+87
115 IF a=0 THEN PLOT x,y: LET x
   1=x: LET y1=y
120 DRAW x-PEEK 23677,y-PEEK 23
   678
125 IF ABS (x-x1)<.1 AND ABS (y
   -y1)<.1 AND a>PI/2 THEN STO
   P
150 LET a=a+s
160 IF INKEY$="" THEN GOTO 100
180 GOTO 20
200 SAVE d1"spirograph"
```

Well, poor old Bob's been in bed with bronchitis; the real nasty hacking sort, and he's just got to the phone to croak to me that he's really got to have what I've typed NOW!! So I guess that's it for this month. Please keep your hints, tips and short items coming to me, John Wase, Green Leys Cottage, Bishampton, Nr Pershore, Worcs WR10 2LX, and I'll try to keep putting a varied and interesting column together.

Oh, and finally....

I went to the Business Show at Earl's Court, last week. Being a good lad, I pre-registered early. In the fullness of time, I had my stuff back to..

Dr. John D. A. Wase, The University of Birmingham, Green Leys Cottage, Bishampton, Nr. Pershore, Worcs WR10 2LX.

So, apart from the fact that they'd got my name screwed up (I'm D. A. John Wase), they also moved the University 30 miles south! And to think I commute 60 miles a day to keep it at arm's length. Full of spleen, I immediately rang them, and demanded that their database be corrected. And sure enough, they did. Too late, though. They'd clearly sold my address to dozens of punters, and the poor postie's been staggering up the drive ever since. Well, at least I know where this one's come from - I've only to look at the address. Just two thoughts. I wonder how many more transmogrifications my name and address will undergo before it finally reaches the Great Database in the Sky-"Can't Deliver; Address incorrect". And if all this can happen to my name and address, I doubt if the Data Protection Registrar really cares about your Speccie. Wicked thought, isn't it!

Right, come on then, contributions urgently required to fill these pages. Send them to:-

John Wase,
Green Leys Cottage,
Bishampton,
Pershore,
Worcs,
WR10 2LX.

See you next month.

MACHINE CODE
WITHOUT THE TEARS

Part 21.

By:- Carol Brooksbank.

This month, we carry on building up our program subroutines. We still have quite a way to go, but at least we have reached the subroutines which do the bulk of the program's work. The first one is the one which draws a block for one enlarged set pixel.

```
BIT_DR   CALL NX_BIT
```

First, NX_BIT is called to move to the correct screen position for the block.

```
BIT_DR2  LD A,(BT)
```

When we are drawing a leftmost pixel of the character (Bit 7) the subroutine is called from here, because there is no need to move along.

```
         LD (BITH),A
         CALL INT_W
         LD (BITW), A
```

The nearest INT(W) is found and stored in BITW as a counter.

```
BT_RPT   LD A,(X)
         LD C,A
         LD A,(Y)
         LD B,A
         CALL PX_ADD
```

The y,x co-ordinates of the starting point are loaded into BC, and PX_ADD is called to find the screen file byte and bit for the point, in HL and A respectively.

```
         EX AF,AF'
         LD A,(BITW)
         LD B,A
         EX AF,AF'
```

The bit number is preserved briefly by using the alternate A register to fetch the counter and put it in B. Another EX AF,AF' gives us A holding the bit number.

```
         CP 7
         JR Z,BIT7
         CP 6
         JR Z,BIT6
         CP 5
         JR Z,BIT5
         CP 4
         JR Z,BIT4
         CP 3
         JR Z,BIT3
         CP 2
         JR Z,BIT2
         CP 1
         JR Z,BIT1
         CP 0
         JR Z,BIT0
```

Now the bit number is checked repeatedly, and when its value is found, the program jumps forward to start setting the block bits at the right x,y co-ordinate.

```
BIT7     SET 7,(HL)
         DEC B
         JR Z,ROW
BIT6     SET 6,(HL)
         DEC B
         JR Z,ROW
BIT5     SET 5,(HL)
         DEC B
         JR Z,ROW
BIT4     SET 4,(HL)
         DEC B
         JR Z,ROW
BIT3     SET 3,(HL)
         DEC B
         JR Z,ROW
BIT2     SET 2,(HL)
         DEC B
```

```
         JR Z,ROW
BIT1     SET 1,(HL)
         DEC B
         JR Z,ROW
BIT0     SET 0,(HL)
         DEC B
         JR Z,ROW
```

The program starts setting bits at the correct one for the x,y co-ordinate. After each one the counter is checked. If all the bits for the width of the block have been set there is a jump forward to move to the next row. If not, the next bit is set.

```
         INC HL
         JR BIT7
```

If bit 0 has been set and there are still some more to do, INC HL moves to the next screenfile byte and we jump back to start setting again at bit 7.

```
ROW      LD A,(BITH)
         DEC A
         JR Z,BIT_EX
         LD (BITH),A
         LD A,(Y)
         DEC A
         LD (Y),A
         JR BT_RPT
```

When a row of bits (the correct width for the block) has been set, the height counter is fetched, decremented and checked. If all the rows for the block have been done we jump forward to exit the subroutine. If there are still rows to do, the height counter is stored again, the y co-ordinate is fetched, lowered by one pixel and stored again, and we jump back to do the next row.

```
BIT_EX   LD A,(Y)
         SUB 1
         LD B,A
         LD A,(HT)
         ADD A,B
         LD (Y),A
         RET
```

Before exiting, the y co-ordinate is restored to its starting position, because it was changed by the subroutine's action.

The next subroutine is the same as the previous one, except that it resets pixels. It is used when the bit we are rescaling was reset.

```
RUBOUT   CALL NX_BIT
RUBOUT2  LD A,(HT)
         LD (BITH),A
         CALL INT_W
         LD (BITW),A
RUB_RPT  LD A,(X)
         LD C,A
         LD A,(Y)
         LD B,A
         CALL PX_ADD
         EX AF,AF'
         LD A,(BITW)
         LD B,A
         EX AF,AF'
         CP 7
         JR Z,RUBIT7
         CP 6
         JR Z,RUBIT6
         CP 5
         JR Z,RUBIT5
         CP 4
         JR Z,RUBIT4
         CP 3
         JR Z,RUBIT3
         CP 2
         JR Z,RUBIT2
         CP 1
         JR Z,RUBIT1
         CP 0
         JR Z,RUBIT0
RUBIT7   RES 7,(HL)
         DEC B
         JR Z,RUBROW
RUBIT6   RES 6,(HL)
         DEC B
         JR Z,RUBROW
RUBIT5   RES 5,(HL)
         DEC B
         JR Z,RUBROW
RUBIT4   RES 4,(HL)
         DEC B
         JR Z,RUBROW
RUBIT3   RES 3,(HL)
         DEC B
         JR Z,RUBROW
RUBIT2   RES 2,(HL)
         DEC B
```

```
         JR Z,RUBROW
RUBIT1   RES 1,(HL)
         DEC B
         JR Z,RUBROW
RUBIT0   RES 0,(HL)
         DEC B
         JR Z,RUBROW
         INC HL
         JR RUBIT7
RUBROW   LD A,(BITH)
         DEC A
         JR Z,RUB_EX
         LD (BITH),A
         LD A,(Y
         DEC A
         LD (Y),A
         JR RUB_RPT
RUB_EX LD A,(Y)
         SUB 1
         LD B,A
         LD A,(HT)
         ADD A,B
         LD (Y),A
         RET
```

Those are the two routines which draw the individual rescaled bits. Now we turn to the one which draws a rescaled byte, by calling them as required.

```
BYT_DR LD A,(X)
         LD (ORIGX),A
         LD A,(VALBYT)
```

We begin by storing the x co-ordinate in another variable. The variable X is changed as the subroutine proceeds, but we need to be able to refer back to the starting position of the byte. The byte we are drawing is fetched from VALBYT in A.

```
MOD1     BIT 7,A
         JR Z,BYT70
         CALL BIT_DR2
         JR BYT6
BYT70    CALL RUBOUT2
```

Bit 7 is tested. If it is 0, we jump forward to a call to RUBOUT, but if it is 1, BIT_DR is called from the point BIT_DR2 - after the call which moves on to the next bit position, because bit 7 is

drawn from the original x co-ordinate. We then jump past the call to RUBOUT, to consider the next bit. All the labels starting MOD are points at which other subroutines may modify this one - in this case, where the "mirror writing" subroutine will modify it so that the bits are checked in reverse order.

```
BYT6     LD A,(VALBYT)
MOD2     BIT 6,A
         LD A,1
         JR Z,BYT60
         CALL BIT_DR
         JR BYT5
BYT60    CALL RUBOUT
```

Bit 6 is considered, and before the call to the appropriate subroutine, the number of bits already done is put in A. This is used to calculate the correct x co-ordinate for the current bit block starting point. BIT_DR and RUBOUT are called from their start from now on, because their calls to NX_BIT can only be bypassed for bit 7.

```
BYT5     LD A,(VALBYT)
MOD3     BIT 5,A
         LD A,2
         JR Z,BYT50
         CALL BIT_DR
         JR BYT4
BYT50    CALL RUBOUT
BYT4     LD A,(VALBYT)
MOD4     BIT 4,A
         LD A,3
         JR Z,BYT40
         CALL BIT_DR
         JR BYT3
BYT40    CALL RUBOUT
BYT3     LD A,(VALBYT)
MOD5     BIT 3,A
         LD A,4
         JR Z,BYT30
         CALL BIT_DR
         JR BYT3
BYT30    CALL RUBOUT
BYT2     LD A,(VALBYT)
MOD6     BIT 2,A
         LD A,5
         JR Z,BYT20
         CALL BIT_DR
```

```
          JR BYT1
BYT20     CALL RUBOUT
BYT1      LD A,(VALBYT)
MOD7      BIT 1,A
          LD A,6
          JR Z,BYT10
          CALL BIT_DR
          JR BYT0
BYT10     CALL RUBOUT
BYT0      LD A,(VALBYT)
MOD8      BIT 0,A
          LD A,7
          JR Z,BYT00
          CALL BIT_DR
          JR BYT_EX
BYT00     CALL RUBOUT
```
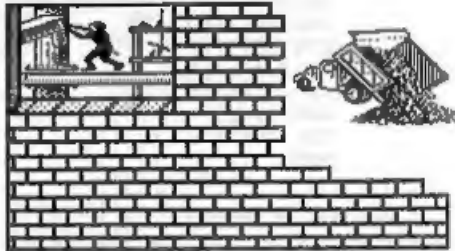
The routine carries on checking each bit of the byte in VALBYT in turn, and calling the appropriate subroutine to draw a rescaled block, always calling them with A holding the number of bits already done.

```
BYT_EX    LD A,(ORIGX)
          LD (X),A
          RET
```

Finally, before exiting, the original x co-ordinate is put back in the X variable.

That is as far as we can go this month. These three subroutines are the longest in the program, but we still have more than two dozen shorter ones to do before we can assemble the code and start looking at the BASIC which will run the program, so I hope you will stick around for a month or two longer!

The HELP PAGE

Edited By:- Kevin Gould.

This month the questions involve quite long answers so we will press on.

First, +3 owner Garry Rowland is having problems with his disc drive. He is certain that the infra-red detector has failed and he wants to know where he can get a replacement. The detector is manufactured by Sharp and has the serial number GP1S52 661.

Garry has been given the telephone number of Sharp's Components Division at Reading (0734 313877). Perhaps he will let me know whether he has been successful.

In view of the scarcity of Spectrum parts and there being only a few firms who carry out repairs it would be useful if any members with recent experience of either of these would drop me a line so that the information can be made available to all the readers.

Roy Burford, thanks for your letter which will appear next month

Now on to printers. Allen Norton of Castleford, West Yorkshire is having difficulty in making the printer control commands for his Citizen 120D printer work with SAM. From the example program sent it is clear that the printer channel has not been set to pass the commands to the printer and only ASCII codes from 32 to 128 are therefore getting through. To set the channel to accept the control codes the instructions CLOSE #3 : OPEN #3;"B" have to be entered. To reset the channel to normal again enter: CLOSE #3: OPEN #3;"P"

Staying with printers.. Tony Jeenes of Malvern has a GLP II printer and would like to reprogram the font to print ½ as a single character (e.g IBM 171) in place of a seldom used character and has read ESC & to define download characters in his manual several times and is still none the wiser (know the feeling well). Ascending and descending characters and the right values for the eleven slots. He has also tried FONT 120.

The manual shows the following format: P1.......... p11 8 down to 1 (CHR$ 27;"&"; CHR$ 0; CHR$ n; CHR$ m; CHR$ A; CHR$ p1; ......; CHR$ p11) N=M for single letter

Hope the following is useful Tony. The matrix you describe is for draught quality download characters. The character you require is plotted on the matrix as per a UDG, however, with download characters you cannot plot pairs of horizontal adjacent dots as the print pins can only strike on alternate columns. The example below gives an example of a simple character on an 11 * 8 grid and the data values which represent that character. The value assigned to each dot of a particular horizontal line is shown down the left side and the 11 data values required are obtained by adding up the values of all dots in each vertical column as shown at the bottom.

Though the matrix is 8 dots high the printer uses 9 vertical pins to print the full range of characters. The terms

ascender and descender describes where the character 'sits' on the print line, an ascender sits on the line and does not use the lowest pin while the latter sits below the line and does not use the top pin.

```
128
 64
 32           .
 16        .     .
  8     .          .
  4   .              .
  2  .                .
  L .    .    .    .    .
    1 2 3 4 5 6 7 8 9 10 11  Data Byte Number
    0 3 4 9 2 3 3 9 4 3  0   ) Value
            6 3 6
```

Please let us know how you get on.

Software is the next subject and we start with E.Warwick of Plymouth. He sent a Spectrum Basic/machine code program which he wishes to convert for use on SAM without having to use an emulator. He converted the Basic part using a translator but cannot get it to run properly because the machine code is still Spectrum based. How can he ammend the program.

The advice is unless you have the following, don't bother! (a) a good working knowledge of programming in machine code (b) posses the "Complete Spectrum ROM Disassembly" (c) the "SAM Technical Manual" (d) Dr Wright's SAM ROM Disassembly discs and finally (e) a good handbook on the way the Spectrum works.

Even if you have all these things, think twice about it as it could cause you a lot of work unless the code is fairly simple.

Do you still want to continue? Yes, then the points to look for are: 1) Does the code make any calls to the Spectrum ROM sub-routines? If such routines make calls or jumps to other ROM sub-routines, manipulate the screen displays or make use of system variables then give up here and now... If the ROM sub-routines are self contained and simple then you can copy them to your own program.

2) Does the code make use of system or program variables? Some system variables can be used on SAM by simply altering the address to that of the SAM variables, others either do not provide precisely the same information or do not exist in SAM. With regard to program variables, the way that these are defined and stored is totally different in SAM thereby requiring an extensive re-write of the code.

3) Does the program use channel information, printer or file routines? These will all need amending.

4) Finally, does the program write/read the display file? If it does then MODE 1 should be selected and the display file which normally resides in the top two pages of memory will have to be paged into the first 64K of memory. If, as it probably will, the code resides above 32K then the display file will have to be paged to sections A & B in place of ROM 0 and the Basic program which might create several problems. Alternatively, the screen routines could be placed in the general purpose buffer space allowing the screen display file to be paged into sections C & D. In either event, this will involve some major modifications to the existing program.

In summary, any attempt to adapt Spectrum code to work on SAM is likely to involve many hours work and quite frankly its easier spending an extra minute or two loading via an emulator

We stay with software for the next couple of questions.

L.J.Shelley of Stourport has Tasword +3 and wants to know how to print boxes around text.

The box characters are found in the IBM character set #2 on most printers and can be used on Tasword +3 subject to two seperate operations being carried out, one to produce the box characters to the screen and the other to send the appropriate codes to the printer.

In order to produce the box characters to the screen it is necessary to know a little about the program. Tasword +3 consists of four program files. These are "RUN", "TC1.BIN", "TC2.BIN", "TC3.BIN" and "TASTABLE.BIN". It is this last file that contains the machine code that produces the letters and graphics for both the 32 and 64 column screens. It loads at 53248 and is 12288 bytes long. The first 2048 bytes make up the letters and graphics for the 64 column screen and the next 1280 bytes do the same for the 32 column screen. These are formed in grids of 4 by 8 pixels for 64 columns and 8 by 8 pixels for 32 columns.

The box characters will be accommodated on the 2nd character set help page and it is first necessary to check which box characters are available on your printer. Some printers contain only those characters which will enable you to print a single line box, whereas others will also include those which allow double lined boxes to be printed.

As Tasword's 2nd character set illustrates some symbols which are not available on most printers it is worth spending some time in deciding which other symbols are to be inserted in the program.

The 2nd character set shows 75 possible characters on the menu but the last six of these are used elsewhere in

the help menu. These are controlled by keys h - m (following symbol shift and s) and are best left unaltered. This will still leave more than enough room for the symbols required for boxes or any of the others you wish to use.

The 64 column 2nd character set starts at 54528 and the 32 column 2nd character set is located at 55808.

The BASIC program "TasChar Editor" (Vol 6 Nº10 Pg 35) is ideal for amending the 2nd character set characters. In fact it can also be used to amend any of the normal 64 column characters which start at 53504. Just load TCE and amend lines 2, 1420, 1520 and 9903. The alternative is to get a supply of graph paper and examine, and then amend the code. It's so much easier using TCE.

It is important at this stage to note what each character was prior to alteration and what the new character is so that the new characters can be inserted into their proper places on the help menu. Save the amended file "TASTABLE.BIN" on a new disc.

Now for the tedious part. The 32 column characters have to be changed and unless someone can rewrite TCE to handle this size it is back to the graph paper. I used DICE when converting my own program and it did make the job a little easier than it would have been if I had had to examine a long list of code.

Transfer the other four parts of the program on to the new disc and load the program. Using the Customisation Menu unlock the help menu and return to text and replace the 2nd character set menu with one containing the new symbols. Go back to the Customisation Menu and lock the help menu (see page 60 of the user manual).

Now for the second operation, the printer codes. Still in the Customisation

Menu you now define the print character translations for each of the new characters (see page 55 of the manual).

Return to text and try out the new characters. If you completely then save the program using the save Tasword option on the main menu.

Mr Shelley also wants to know how to print text files from Tasword 2 when using the +3.

First, load TW2 and go to Basic. Edit line 250 by deleting RANDOMIZE USR (FN p(62472)). Next, edit line 15 by inserting at the beginning of the line:-
FORMAT LPRINT "U": LPRINT CHR$ (17)
Finally, return to the main menu and save Tasword on a new disc.

Just before closing, we can only accept problems in writing, give as much detail as you can, the more details we have the better we can answer your question.

My continued thanks to the Team for their help and assistance and we will be back next month.

As usual send your problems/answers to the following addresses:-

Anything not SAM or +3.-
Kevin Gould (Format Help Page),
2, Barleyfield Close,
Heighington,
Lincolnshire, LN4 1TX .
Anything SAM.-
Ray Bray (Format Help Page),
'Elmsleigh',
4, Tidworth Road,
Porton,
Salisbury,
Wiltshire, SP4 0NG.

Anything +3. CP/M.-
Mike Atkins (Format Help Page),
70, Rudgwick Drive,
Bury,
Lancashire, BL8 1YE.

WHAT THE HECKs This All About?

By:- Nev Young.

How can 9 + 8 = 11?
Why is 1.50 + 0.40 = 2.30?

Its alright. I've not gone mad, both the above are true. There just happens to be some extra information missing. The first sum is in hexadecimal, the second hours and minutes.

The reason for this is that many many people just do not seem able to grasp hexadecimal notation and what it is about. But as the second example shows we all manage to use duodecimal and sexagesimal with no problems. I hope in this article to explain what hexadecimal is, how to use it, and above all why bother!

Many years ago when computers were big and unreliable all those marry people who had to put data into the computers were told, quite rightly, that they had to learn the computers way of working as the computer did not have enough power to be taught how we worked. This meant all data was entered, and much printed out, in BINARY. Long strings of zeros and ones. Here we get the first piece of modern day confusion.

True/false, on/off, 0/1 These terms are often used and exchanged freely but the reason not explained. It began with the simple switching circuits in the early machines. Any circuit could be on or it could be off, there is no middle ground of nearly on, or not quite off. It was on or off. Just like a light bulb in a room ON or OFF. If the light is on then 'it is true that the light is on', if the light is off then 'it is false that the light is on', That should explain true/false and on/off but what about this 0/1 stuff?

A computer with only one switch will not do much. It can't even be called a computer. So lets make a bigger one with five switches. Each can be on or off. If we place them in a row (just to be tidy) then we can read of the state of each 'on on off off on'.

Not too bad for five, but what about 5000? People and other human beings are generally lazy and so a short way of writing this was found. An obvious choice would be the initial letter of the works on and off (Well it would be in French, a and e). Think Again.

Let us think about why we want to talk about these circuits being on or off. There are two very fundamental things that computers do. One is decide to do or not do something depending on whether something is true or false (its those words again). The other thing is to add numbers together.

True and false is fine. That's just a circuit being on or off. Adding up is somewhat different. As a circuit can only be on or off then one circuit can only represent a number or the absence of a number. Now it may come as no surprise to learn that binary arithmetic was invented many years before computers and the rules were well defined and it works. If you are not sure what binary is then let me refresh your memory. It is a

number system based on the number 2. So the only numbers you can use are 0 and 1 (now where have I heard that before).

The rules for binary arithmetic are very simple: 0+0=0, 0+1=1, 1+1=10.

So to get back to the computer circuits, on and off could be used to represent 1 and 0. Now we can add up numbers. Lets take the five circuits we had before 'on on off off on' these can now be written as a binary number 11001, great! Lets think of another number 01101 and add then together 11001 + 01101 = 100110

Is anybody lost? OK then step by step:-

```
11001      11001      11001
01101      01101      01101
    0         10        101
  1
1+1=10     0+0+1=1    0+1=1

11001      11001      11001
01101      01101      01101
 0101      00101     100101
1          1
1+1=10     1+0+1=10   0+0+1=1
```

Still lost? Ok then give up.

So we now know why on and off are sometimes true and false and why sometimes they are 1 and 0. (I hope). Things are not too bad for our example of 5 and 6 bit numbers (I say bit because I do not want you to think that 0 and 1 are digits like 7 8 9. It is also their correct name).

Now what about BIG numbers. For example the address of the start of the screen on a 48K Spectrum is 0100000000000000. Do you agree that is a bit hard to see and take in all at once? I think so, so lets split it up into groups, I think 4 to a group should look OK. 0100 0000 0000 0000. Yes much better. That's just the same as we do in decimal with numbers like one hundred thousand

million (100,000,000,000) except we group the digits into threes.

I can almost hear you say "When is he going to get to hexadecimal?" Do not fear, almost there.

By the time binary numbers were being written in groups of four (there had been a period of grouping them in threes) computers had got a lot smarter and smaller (but no more reliable). The people who used them were getting very tired of writing all those zeros and ones and wanted a quicker way of writing it. Then somebody had a think and saw that in any group of four bits there are 16 different bit patterns.

All we humans had to do was think up good meaningful names for each of the 16 patterns and we would have a quick way of naming the big binary numbers. The names used are:-

```
0000 zero
0001 one
0010 two
0011 three
0100 four
0101 five
0110 six
0111 seven
1000 eight
1001 nine
1010 ten
1011 eleven
1100 twelve
1101 thirteen
1110 fourteen
1111 fifteen
```

Not very imaginative, and not a good choice either. People being people wrote the binary groups of numbers using the decimal representation of their names. That is for 0110 they wrote 6, for 1101 they wrote 13. Now I can handle that, when I was a lad I was taught that there are 16 ounces in a pound. So for 1001 + 1000 = 1 0001 people would write 9 + 8 =1 1. Now there is the problem. If you miss out the space in the answer 1 1

becomes 11. This is wrong. It is bad. It is confusing.

So somebody had another think. This time it was decided that the binary patterns would have different names:-

```
0000 zero
0001 one
0010 two
0011 three
0100 four
...
1001 nine
1010 A
1011 ■
1100 C
1101 D
1110 ■
1111 ■
```

A bit of a compromise if you ask me but we are now stuck with it. This is known as hexadecimal notation. Where every group of 4 binary bits is written as a single hex digit 0123456789ABCDEF

So now 9+8=11 is OK because 11 is not the same as B.

No sooner had this been sorted out and agreed than computers became so small, powerful and unreliable that we humans could use our own number system, decimal, and let the computer convert it into binary and back into decimal for us. So nobody really needs to use hex any more. So why does it persist?

You remember the other thing computers do, decide to do or not to do something if a circuit is true of false. Well just as we grouped lots of bits to make big binary numbers it was easy to group lots of trues and falses together, and in the same way as we wrote the binary numbers (0100 0000 0000 0000) then the groups of conditions were also grouped (FTFF FFFF FFFF FFFF). And in the same way this was boring to write so it got shortened and hex was ready and waiting for it.

Just imagine a group of conditions

FTFT TTFF this would be written as 5C. Now what if something should happen if the fifth bit from the left is true. First we have to get rid of all the other bits so we only have the one we are interested in.

This is done with a LOGICAL operation, in this case AND (written as &). The rules are F&F=F, F&T=F T&T=T. So to get only the fifth bit:-

```
   FTFT TTFF   or in hex   5C
&  FFFF TFFF             & 04
=  FFFF TFFF             = 04
```

You see it is easy to 'see' the bit patterns from hex, whereas if you used decimal you would have 172 & 16 = 16. So hex is often used when the BINARY pattern of the group of bits is more important than the value. That is why hex is still used.

A few examples:-

```
hex    binary    decimal
Hex 5C >> = 2E
Binary 0101 1100 >> = 0010 1110
Decimal 176 >> = 78

Hex 5C & 42 = 40
Bin 0101 1100 & 0100 0010
               = 0100 0000
Dec 176 & 66 = 64

Hex 5C ; 42 = 5E
Bin 0101 1100 ; 0100 0010
               = 0101 1110
Dec 176 ; 66 = 178
```

(Note for SAM users & is BAND and ; is BOR)

It makes sense, to me at least, that if you are discussing something that requires the use of one number system then you should use that number system throughout to avoid confusion. Otherwise if I gave you the number 64 is that decimal 0x0 or hex AxA?

Even if I have only managed to confuse the issue further I hope that I have taken a little of the mystery away from

hex and maybe even our editor will allow its use in his magazine.

EDITOR'S NOTE:

Contrary to popular belief I am not opposed to Hex. I just like to keep it in it's place. For ANDing and ORing, masking bits etc, it is good (at times) although I feel that in the above examples the Binary version is much easier to understand.

For most addresses, data values and other numbers it is always better to use decimal notation. After all, the school system takes many years to knock base ten arithmetic into people- why not let the computer do all the work.

So, yes, it is worth getting to grips with Hex - it comes in useful once in a while and you shouldn't be afraid of it. But far more people will learn machine code if we leave out all the un-necessary Hex.

What do you the readers think? Write in and air your views on Hex (and other number bases).

Mum, I've got a little gremlin into trouble!

YOUR LETTERS

Dear Editor,

I am writing to offer my services to your forthcoming Games Supplement. I would love to contribute to this very welcome new publication by writing games reviews for you. In fact, if you are still looking for an editor, I would be very interested.

I have games reviewing experience. Last summer, I had the opportunity of working on the sadly now defunct *Sinclair User* magazine. Over the September, October and November issues (numbers 127-129), I wrote a great number of games reviews, wrote instructions for the 'Great Eight' pages, single-handedly produced a two-part pull-out tutorial for the art utility 'Icon Graphix' and even edited and answered the letters pages in issue 128. But that's enough bragging.

Yes, I am only sixteen, but don't let that perturb you! I have sent a reference from *SU's* old editor, Alan Dykes, although he got my name slightly wrong (not surprising seeing as he wrote it two months after I had left). I could be free to supply you with a monthly batch of reviews etc., (in between doing my A-Levels). If you decide I can be of use please get in contact and we'll start to plan in more detail.

As you said in the April edition of your meritorious magazine, the launch of *FORMAT's* own games mini-mag is a much needed venture now that *SU* has departed, it will certainly compliment your more programming and technically-orientated main publication, meaning *FORMAT* will now cover all aspects of the Speccy/SAM market, and therefore confirm it's position as the best magazine available for these wonderful computers.

Thank you very much,

Yours sincerely, Marc D.Richards.

I'm still looking for an Editor (if you lived a little nearer you would do by the sound of it) to handle *Games Format* - like the title? Jus' came to me that did.

Still, thanks for the interest Marc. Will be in contact. *Ed.*

Dear Editor,

I have been a subscriber to *FORMAT* Magazine for more than six months and a SAM owner for more than 18 months. I would like to know about some certain items which used to be available for the Sam.

Firstly, while looking through some of my previous issues of *FORMAT* I noticed on offer on Blue Alpha's Voice Box. I would like to know if it is still possible to get hold of the Voice Box elsewhere?

I would also like to know what happened to Blue Alpha's Sound Sampler. I have not seen a mention of this since the fall of SAMCo. That also goes for SAMCo's wonder of 30,000 colours, Kaleidoscope. Am I to assume that these are gone forever? Also, what has happened to the Spellmaster which was set to revolutionize SAM word processing.

Finally, I would like to know if we are ever going to see the video digitiser again. I would hate to think that it has been forgotten about. I hope you will be able to enlighten me on these items. I have never owned any of them myself but I would certainly consider buying some of them if they were still available.

Yours sincerely, K.J.Mayes.

Both the Blue Alpha interfaces are now out of production but Kaleidoscope will be available again soon. The Video Digitiser will appear eventually, I'm sure, just keep reading **FORMAT** and we will let you know when it comes out.

Spellmaster is still selling, all you need to do is contact *FRED*, why did you think it had disappeared? Not everything can be advertised all the time you know. *Ed.*

Dear Editor,

Thank you for publishing my letter in the August issue of *FORMAT*, especially your comments about a 128K emulator.

You asked "Why bother?" Well, I seriously think it would be worth it to allow users such as myself to run 128K specific software, some of which was truly 128K - rather than just 48K with a tune tacked on - and to allow SAM users access to software that gives a better idea of what Spectrum software could be like given the right amount of memory. There seems to be a constant stream of complaints from the SAM owning fraternity that all they seem to have released for the Coupé are puzzle games. If SAM owners were able to play software like 'Carrier Command', 'Where Time Stood Still', 'PANG' and 'Gauntlet 3', and were able to play many games without having to use the 48k Multi-load, I think they would be quite happy to wait until some software that really stretches SAM is released. The

SAM really is a superb machine with the capabilities to match the 16-bit machines, as 'Prince Of Persia' and the soon-to-be-released 'Lemmings' show.

I know that most of your readers are 'serious' owners, and my comments may seem games orientated, but there's also the 128k versions of 'Tasword', 'Art Studio', and 'Icon graphics' to consider. I realise that a 128k emulator does pose serious hardware problems with SAM, but I'm sure someone somewhere could produce a 'Dangle' that would do the job. I would buy a 512k SAM like a shot if I thought I could still play my 128k only software, and use 'Art Studio' in its extended form, or 'Tasword 128'.

One final point, I have managed to solve my incompatible joy-sticks problem - up to a point. I have managed to buy a pair of '+2 joy-stick leads' from a mail order company called 'Software City' These leads allowed me to use a pair of Quickshot 2 turbo joy-sticks directly into the +2A's joy-stick ports. Not ideal, as only the Interface 2 is supported, but there are only a few titles that I have that don't support that standard.

By the way, any news on the planned games magazine, as I'm chomping at the bit, ready to contribute in any way I can.

Yours sincerely, Jon Rose.

I've never heard any complaints about Puzzle games on SAM - except from the crowd at *YS* who always seemed to find a puzzle in every game they tried.

As for the 128K software - I'm only partly convinced. I think serious software is now well covered on SAM, so it would only be to avoid multi-loaders that most would go for a 128K emulator.

Still, if the right hardware man got together with the right software man, then it may be possible - at a price. *Ed.*

Dear Editor,

I see that you are working on an improved User's Manual for SAM - good news! If you need any 'betatesters' I should be very happy to oblige. I exchanged some views with Andy Wright when he was thinking of writing one, and would dearly love to see a proper guide.

When I first learned to program (nearly 40 years ago), the importance of good program instructions was dinned into us, it was recommended that there should be two parts: a 'user guide', followed by a precise specification of what the program actually did. This, of course, is more or less what the ZX81 and the 48K Spectrum Manuals did, with Appendix C acting as the specification. The slim manual for the Spectrum+ was a bit of a drop-off, with nothing, for example, about priorities, or system variables; but its 'Reference Guide' was pretty good, as far as it went. In particular, it made a good job of specifying syntax (which it called 'Format'), except that the omission of priority-rules meant it was incomplete.

So, full rules of syntax is one of the things I should like to see; the other is more precision in specifying what happens. A good example is BRIGHT, which just recently I had occasion to look into. The Guide's description of its effect is not good at all, for (in MODE 4 at any rate) BRIGHT 1 replaces the current palette position p by p+(8 AND p<8), and BRIGHT 0 by p-(8 AND p>7). Such precise information would enable a serious programmer to use BRIGHT in full knowledge of its effect.

I look forward to your new User's Guide.

Yours sincerely, Ettrick Thomson.

I really lumbered myself when I agreed to take on the task of editing a new SAM manual, it is going to take some time and I can do with any help I can get.

By the way the article you sent will appear soon, I'm just a bit behind due to converting to our new DTP system. *Ed.*

Dear Editor,

Thank you for publishing my letter in last month's FORMAT. It will be interesting to see whether I get any response on 'C' and/or the 128K keypad.

Regrettably, I cannot also thank you for your response to the letter from Geoff Winkless.

In my view, your comments were both childish and grossly offensive, and I hope that on reflection you might feel that an apology to Geoff would be appropriate in the next issue.

I do not know whether you are familiar with KEdisk or with Geoff's demo work. KEdisk is a very useful suite of programs, primarily aimed at SAM/PC interchange but with an excellent sector editor (comparable to that in the original NORTON programs for the PC) that you can use purely with SAM disks, which I use frequently. You should not deride demos, since they are excellent publicity for SAM showing, as the good ones do, that SAM is the graphical equal of some more exotic machines.

The point is that in my view SAM needs all the help and support it can get at the moment - the whole scene seems to be shrinking, with magazines disappearing, not much serious software coming on the market and people like Geoff deserting what they obviously see as a sinking ship. Responses like yours can only serve to make things worse.

What you should have done was to encourage him to continue to program

for SAM, even if his main interest is to tackle other machines that apparently are more suited to his present needs.

Yours sincerely, Alan D.Cox.

I will leave out most of what I was going to say in reply to your letter Alan because the next two cover most of the points. But I will say that it was Geoff Winkless who was being childish - not (I hope) me. *Ed.*

Dear Editor,

Have just received my copy of Vol.7 Nº1, the September issue of FORMAT - what a super innovation with this month's COVER!? I can now take my FORMAT to work and handle it with impunity, knowing that should the cover become soiled, all I have to do is to remove it to reveal the spare one beneath! Brilliant - look forward to all future issues being the same?

I must say that I absolutely agreed with your comments with reference to Geoff Winkless's letter.

Whatever shortcomings the SAM may have in his eyes, as far as I am concerned, the SAM is an excellent utility machine. I use my SAM for my hobby of Short Wave Radio Listening, which with the appropriate software allows me to decode MORSE CODE - FAX -RITTY - SSTV - digital pictures etc etc., Using FLASH and, with Steve Nutting's mini Dump, I design my 'Short wave reception report' letter heads etc., and I am typing this letter using 'THE SECRETARY' - and for my Data bases what better can I have than Steve's SC_FILER program.

Now about games - well when I have a few idle moments I run up the Speccy version of SUPER CHESS.3 probably the best ever? or have a game of SCRABBLE! I have played a few games

on the Sega and Nintendo machines, and must admit the super graphics, but quite honestly you play one you've played the lot. How about you programmers out there coming up with a SUPER ORTHELLO? The SAM version I have is easy to beat.

Quite honestly, with a 512k SAM, fitted with Steve's Auto Load ROM chip - 2 Drives - SAMBUS & 24 pin LC24-10 STAR Printer, what more can I want? It does everything that I require, surely it's not just all about playing games!? although one or two good simulations as mentioned would be welcome.

Should any SAM user interested in Amateur Radio require the above mentioned decoding software write to:-
Ian Spencer (DJOHF/G3ULO) Fichtenweg 1OC, D-53804 Much, GERMANY.

Finally, I must thank you and all your contributors for an excellent magazine- and also my sincere appreciation to poor OVERWORKED and UNDERPAID Jenny, for her undying dedication in getting our issues out to us on time - thank you.

Yours sincerely, Ron Deeks.

Thank you Ron for your appreciation. Mr James Reed from Kent recently wrote on his renewal notice, in reply to the question 'what would you most like to see covered in a future issue?', his answer was 'Any Amateur Radio Buffs out there?' Well James, now you know.

Oh, yes, the double cover on your issue was just a one-off accident - otherwise Bob's printing bill would go up. *JENNY.*

Dear Editor,

Last month you printed a letter from Geoff Winkless. I can't express in words the anger I felt. How can he complain that the SAM is too slow and can't

compete with the Atari Falcon? It wasn't meant to compete. You don't buy a mini and expect to beat a Formula 1 car in a race.

If the SAM is so terrible, why did Chris White write POP and then immediately start Lemmings?

It is also utterly wrong for him to turn round and start slagging off perfectly good games (which the vast majority of SAM owners enjoyed) just because he isn't man enough and isn't making enough money. I don't expect anyone with any sense has any reason to be this spiteful.

I program utilities and games for the PD scene, and enjoy what I do and don't get a ha-penny. For his work on just ONE PROGRAM he made £150 and he still moans, and as for the demos, Mr Brenchley is quite right in saying that we don't want more demos.

His comments that an accelerator is the only way to make SAM a *real* computer is utter garbage. The Spectrum only had 4Mhz yet was, and still is used widely and is still a good computer - its not how many bits it can handle, but how it handles them.

Most of us are very happy with E-Tracker and have no real need for samples.

Overall Mr Winkless, we are better off without you and I can't see why you are even trying to complete your current project because I for one certainly won't touch it with a barge pole now.

I'm sorry about this, he made me so angry and I had to get it off my chest.

PS. Happy Birthday Format.

PPS. Did you receive my ticket order for the Gloucester show? My friend and I can hardly wait!

Yours sincerely, Andrew Chandler.

Thanks Andy, well put. Your letter covers quite well the comments made by several others this month.

Tickets for the show go out soon. *Ed.*

Dear Editor,

I would be very grateful if you could inform me when my subscription runs out and details of how to renew.

Further to the letter appearing in Vol 7 Nº1 mentioning the quarterly Si-Fi mag, as I have said before, in recent correspondence, this is a fantastic idea and I would like to help if I can.

Up the SAM and down the Amiga (even though I own one).

Yours sincerely, T.J.Cooley.

Well it has been some time since I last told people I guess...

Look at the top line of the label used to address your *FORMAT* to you. The first five digit number is your membership number, that is followed by your expiry date and then a sort code we use. Let's say the expiry date is 0394, that means the membership runs out at the end of March next year and the last issue will receive will be April 1994. But don't worry, your last two issues will contain renewal notices - all part of the service.

I'm still looking for an editor for *VOID* (and for the games supplement). *Ed.*

Dear Editor,

Thank you for another 12 months of **FORMAT**.

I have a great interest in SAM Machine Code, and could write for you articles on adding new commands to SAM Basic and manipulating MODE 3/4 screens. Please let me know if you would be interested.

Yours sincerely, Tim Wells.

I would love to see anything you can

produce Tim, it is always good to see new writers. Give me a ring if you need any guidance. *Ed.*

Dear Editor,

My father-in-law is an enthusiastic collector of data on his family tree and is a member of the 'Single Name Society'.

Upon his death (which hopefully will not be for many years!) he wishes to leave his data to the society.

The problem is that all his data is on a SAM - but the Society's computer standard is IBM PC.

Is there any way we can down load files created on the SAM with OUTWRITE! and SC_FILER onto a PC running MS DOS 6.2?

See attached note for details of his hardware configuration.

Yours sincerely, Peter Frampton.

These IBMs get everywhere don't they, and software to run on them to load alien disc formats is %*#$ expensive.

Still, help is at hand from SD Software. Their PC SUITE will copy data files to (and from) 720K PC discs. Most database programs on IBM compatible machines will import ASCII files so provided the data is in a known format there should be no problems.

PC SUITE also opens the possibility of your father-in-law getting data into his SAM from other society members. *Ed.*

Dear Editor,

Wow! Yet another great FORMAT magazine ......! The new typeface you use makes it a lot more pleasant to read. Keep it up!

This first question would be more appropriate if it was sent to the Help page. It's such a small question that I didn't want to bother the Help page or the Hotline. What does the "83

Statement End Error" mean on the SAM? So far, I've managed to narrow it down to when the computer is READing some DATA. Rewriting the whole DATA line from scratch was the only remedy I could find to solve the problem. I don't even know what was wrong with it. Any Ideas?

I know the reason for my lack of programs for the Short Spot section...... Lack of inspiration. Just as artists need it for their work .... we programmers need it for ours. That challenge in the Short Spot section was intriguing. Keep that up, please.

On the subject of Short Spot, John Wase asked us to report on how fax paper would work for the Brother thermal printer. I would like to tell him that I use the Brother HR5 Thermal Transfer printer with common everyday 'fax' paper and it works lovely. The trouble is that the roll is too big for the printer's own roll holder so you will need to attach a bigger roll holder to your desk.

I should go now. Long Live FORMAT!

Yours sincerely, Justin Skists.

Error 83 usually means that the line has become corrupted - Basic can't find a sensible end to that line. What causes it? Your guess is as good as mine - all I can say is that it rarely happens to me.

SD's PC Suite will transfer files to and from PC (see an earlier letter this month). *Ed.*

Letters may be edited or shortened to fit on the pages.

PC=Z80
CONVERT YOUR PC INTO A SPECTRUM

By:- Stephen Baines.

An essential for any Spectrum user moving onto the PC, with little or no knowledge of a PC, is a Spectrum Emulator. A wide number are now available on the PC platform, and Z80 by Gerton Lunter has built up quite a following since its initial release.

I originally bought v1.45 last October, and had used it frequently since I bought a PC in December. I didn't expect such a good package, and I was impressed with its emulation of the 48k Spectrum. It works on what is now considered a modest PC (a 386SX 20MHz), and unlike some other emulators (such as JPP) it some emulators (such as JPP) it will even run on lesser systems, albeit rather slower. Another plus point is that it will also use your DISCiPLE/PLUS D format discs of software. Snapshots of your games, and other software are converted into a format known as the Z80 format. These can be loaded into the emulated Spectrum.

The program is now at version 2.01, and the changes are major to say the least. On my PC it will run at 93% of the speed of a normal Spectrum, a hardly noticeable difference to a proper Spectrum, and no change on the version 1.45. The minimum system required is 512k memory. The choice of video adapters is impressive (VGA, EGA, Hercules, CGA or Plantronics). If available (and really an essential for using all the 128K features) EMS memory, an Adlib sound card, and an analogue or digital joy-stick. The major

change since the last version is that it will now emulate a 128K Spectrum in all its splendour. It also emulates the Interface 1 accurately, with up to eight Microdrives attached to it. Machine code programs will also communicate with it as it emulates both the legal hook codes (as suggested by Sir Clive) and the illegal calls as used by many programmers. This means that any of your old software that you converted from using Microdrives to run with your PLUS D or DISCiPLE may need a slight re-write (or use the original version).

If you have an ADLIB or compatible card, the emulator will even emulate the 128's sound capabilities. It even emulates the notes in a far superior quality to the original. You see, not everything can be totally emulated. Noise channels are difficult. As a result games such as 'Nigel Mansells Grand Prix Simulator' (Martech circa 1988) sound very odd indeed. Other games relying on tunes, such as Rainbow Islands, run much better. Joy-stick emulation is fully featured. Any joy-stick can be emulated in such as Kempston, Sinclair and Cursor joy-sticks.

The other Spectrum Emulators also have snapshot facilities. Unfortunately, not all emulators use the same format, and so a conversion routine is required. This can be done outside the emulator, although some formats (*.SNA) can be read directly.

Your old basic and machine code

programs can also be converted to run on the emulated Spectrum. This is done, again by the versatile DISCiPLE disc reader, they are converted to what are known at TAP files (Tape emulation). These can be loaded with the familiar LOAD"name".

You can run the emulator in a DOS Box under Windows. This is not recommended on anything less than a 486 with lots of RAM. On my 386SX 20MHz, with 2MB of memory, it is unbelievably slow. It's always a good idea to leave Windows and run the software in its native environment. It also lessens the risks of hanging your system.

Another odd facility is that you can run non-standard ROMs. This may seem very odd, but it gives you the opportunity to fiddle with the Spectrum ROM, have your name on the power up, or write your own ROM from scratch. Also, you can use other Spectrum clone ROMs, such as the Microdigital clone, or if you can find one the Timex 2000.

There are lots of options available to tweak the Spectrum. Most of these you will never have to touch, unless you come across a particularly finicky (and generally older) program. 48K mode may seem very odd to work in, unless you know all the key positions by heart, although help is at hand with a Hot Key that puts up all the keys as they are on the PC. As a rule though it is easier to program in the 128 mode, as you can just type in the program, and then switch to 48k mode.

Do you hanker for your old tape player? Do you wish you could fiddle with the loading level for old times sake, or for those programs that didn't like the PLUS D? Yes, then make the tape interface. That's right, you too can run a

tape player into your PC from the parallel printer port. This works rather well, and is a good way of transferring all your old programs into the PC if you couldn't convert them before. There are the odd exceptions, for example, no matter how I tried on my computer, or my brothers 486, it was impossible to get 'Bubble Bobble' to load, the turbo loader used appears to be rather unusual by Spectrum standards and just would not load the game. On the whole though tape interface works pretty good, and the loading and saving is implemented very well. There is even a utility to transfer your programs back onto tape again (compressed for re-loading onto a normal Spectrum), or you can save them to a proper tape anyway. The interface is very easy to build, it has only a few resistors, 2 capacitors and the connectors. No IC's or transistors to melt, and anyone half-handy with a soldering iron, a Vero Bloc re-usable board, or a Vero Wire Prototyping pen (as I used - no risk at all of melting anything), can easily manage the job. It costs less than £5, and makes the whole package all the more worthwhile.

There is an addition that will puzzle most *FORMAT* readers. It has a facility called SAM RAM. No, this is not a clone of the SAM computer. It was originally a hardware device made for the 48k Spectrum, and is in effect a good equivalent of a Multiface, with a powerful monitor. It works very well, and allows the use of many of the published Multiface pokes as well as allowing you to look at how programs work.

One small grouch, for anyone who uses the Tasword Plus 2 program with the PLUS D/DISCiPLE conversion, the disc reader won't read opentype files. You

need to reload them into your original Spectrum, and then save them to tape, and then re-load them from tape to the PC. There is an alternative, you can build a RS232 connector for Spectrum to PC. This will allow you to send the programs down a wire, like a network. The manual gives all the data you need for building all the bits, and is really easy to follow. I haven't tried yet, as I am waiting for a part, and I am inept at wiring Telephone-type plugs.

Screen shots and text files can be converted into PCX & GIF files (pictures) or ASCII-type files (text). This makes writing manuals for your own software very easy, as screenshots can easily be incorporated.

On the whole the emulator works very well. You really wouldn't notice you weren't running a real Spectrum. Compatibility is very impressive, with the odd exception, and software such as this is a must for anyone moving to the PC. Also, emulators of a quality such as this deserve to be supported to keep the Spectrum alive.

Now down to brass tacks. The cost? It is extremely modest. It is released under the Shareware concept. This means that for little cost a demo version (with tape support disabled, and none of the disc readers) can be bought to try out. This costs £2.50 from B.G.Services in the UK. The full version is extremely good value at £15 (£10 to those upgrading from an older version). This is an essential addition for any user of a PC who occasionally wishes to run their old programs, or wants to program in something a little more friendly than QBasic.

Contact:- B.G.Services, 64, Roebuck Road, Chessington, Surrey, KT9 1JX.